Probabilistic Reasoning in Complex Dynamic Systems

Avi Pfeffer
Harvard University

Objective:

Develop methods for reasoning probabilistically about complex, time-varying hybrid systems.

Design algorithms for diagnosis, prediction, and real-time monitoring.

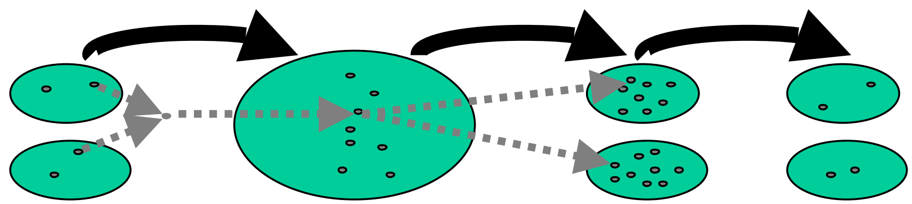
Key Innovation:

Exploit hierarchical decomposition of complex system into weakly interacting subsystems.

Accomplishments So Far:

Developed and implemented factored particles algorithm for efficient monitoring. Experiments show it to be more effective than existing algorithms. Paper in UAI-02.

Tasks:



Design and implement hybrid, hierarchical particle filtering algorithm for flexible, efficient reasoning [DONE].

Dynamically allocate computing resources to subsystems based on need.

Determine appropriate time granularity for reasoning in subsystems.

NASA Applications:

Advanced life support, on-board health maintenance, failure diagnosis

Participants:

Avi Pfeffer, Leonid Peshkin, Brenda Ng

Complex Stochastic Systems

- **dynamic:** system evolves over time
- **uncertain:** non-deterministic transitions, exogenous events, noisy sensors
- **hybrid:** discrete and continuous variables
- **complex:** many interacting subsystems, each with many variables

exploit *hierarchical decomposition* into *weakly interacting subsystems*

Reasoning Tasks

- **diagnosis:** determining likely sequence of past events that lead to observations
  - discover causes of mission failures
  - help determine maintenance strategy
- **prediction:** determining whether an event is likely to happen in future
  - determine value of control actions
  - predict mission success
- **monitoring:** maintaining beliefs about state of system in real time
  - prerequisite for online control

Monitoring Applications

- Robosphere
  - many interacting subsystems, e.g.
    - temperature
    - electrolysis
    - plant growth
- Mars rover

Real-Time Monitoring Step

- Given:
  - a distribution over state of system at time t-1
  - model of system dynamics
  - observations at time t
- Produce distribution over state of system at time t
  - *belief state*

Challenge of Monitoring

- Exact monitoring intractable
  - cannot represent full joint distribution over state space
- Need approximate algorithms that are real-time fast and reasonably accurate
- Two approaches:
  - sampling (particle filtering)
  - parametric (Boyen-Koller)

Particle Filtering

- Belief state represented by set of samples
  - each sample is possible state of the world
- Algorithm propagates samples from t to t+1 by importance sampling

+ Convenient and flexible
+ More samples: better results
+ Converges to correct answer

− High variance
  - especially in high dimensional spaces
  - need too many samples for reasonable accuracy

Boyen-Koller

- Assume state is described by set of variables
  - as in Dynamic Bayesian Network
- Divide variables into clusters
- Maintain distributions over each cluster instead of over full space
  - as if clusters were independent

\+ Exploits system hierarchy

\+ Works very well when it works

\- Inflexible
  - breaks down for very large models

Our Approach: Factored Particles

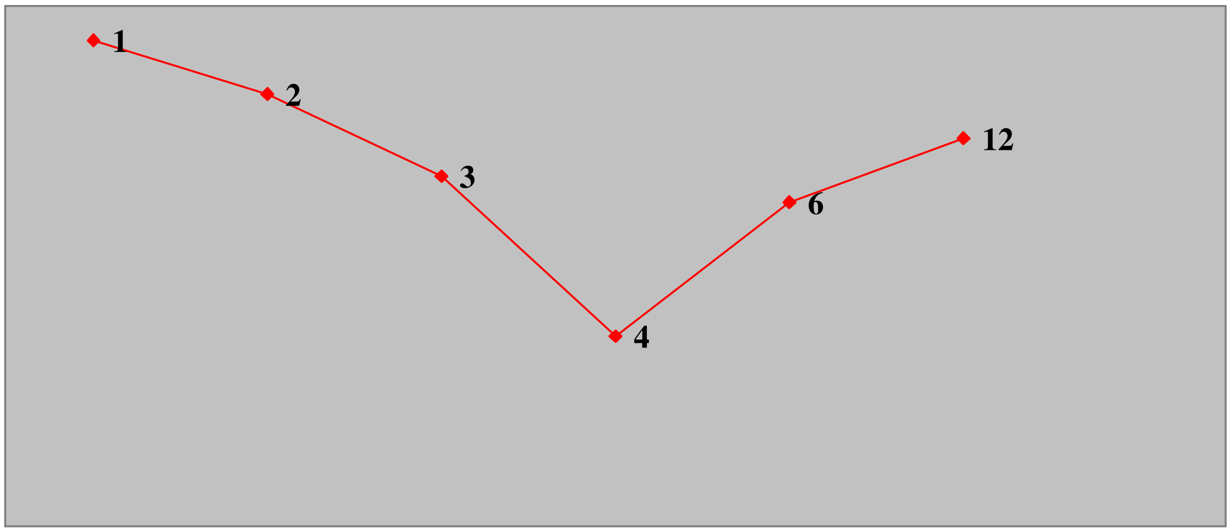
- Combines flexibility of particle filtering with exploitation of system hierarchy
- Divide variables into clusters, as in Boyen-Koller
- Maintain samples of cluster states, instead of samples of complete states
- Three methods of propagating cluster samples
  - see UAI-02 paper for details

Why it Works

- Approximation 1: assume true distribution is product of cluster distributions
- Approximation 2: represent each cluster distribution as set of samples
- Two approximations are better than one!
- Unlike Boyen-Koller, we can propagate cluster distributions in large models
- Much lower variance than particle filtering
  - samples are over low-dimensional spaces

Results

- Across-the-board improvement over particle filtering in medium to large problems
  - up to 50% reduction in error (same running time)
- There is an optimal number of clusters
  - bias-variance tradeoff



Identifying Clusters

- Can we automatically determine
  - how many clusters to use
  - how to assign variables to clusters
- Tools for the task
  - clustering algorithms from machine learning
  - sensitivity analysis
  - separability and near-separability

Dynamic Allocation of Resources

- Can we dynamically allocate a different number of samples to each cluster, so as to maximize use of computational resources
- Need
  - an extended inference algorithm
  - a way to determine how many samples to allocate to each cluster
    - may be online or offline

Other Future Work

- Extend algorithm to hybrid systems with discrete and continuous variables
- Extend algorithm to subsystems using different time granularities